09:59:03 OCA PAD AMENDMENT - PROJECT HEADER INFORMATION 06/16/95

Active

Project #: E-24-X50 Cost share #: Rev #: 3
Center # : 10/24-6-R8117-0A0 Center shr #: OCA file #:
Work type : RES
Contract#: 94-G-016 Mod #: 01-ADMIN Document : GRANT
Prime #: Contract entity: GTRC

Subprojects ? : N CFDA: 20.108
Main project #: PE #:

Project unit: ISYE Unit code: 02.010.124
Project director(s):
NEMHAUSER G L ISYE (404)894-2306

Sponsor/division names: US DEPT OF TRANSPORTATION / FED AVIATION ADMIN
Sponsor/division codes: 124 / 008

Award period: 940501 to 951031 (performance) 960131 (reports)

| Sponsor amount | New this change | Total to date |
|---|---|---|
| Contract value | 0.00 | 53,294.00 |
| Funded | 0.00 | 53,294.00 |
| Cost sharing amount | | 0.00 |

Does subcontracting plan apply ?: N

Title: AIR TRAFFIC CONGESTION DELAY OPTIMIZATION

PROJECT ADMINISTRATION DATA

OCA contact: E. Faith Gleason 894-4820

Sponsor technical contact

TOM MIFFLIN, AOR-200
(202)287-8525

FAA NATIONAL HEADQUARTERS
800 INDEPENDENCE AVE. S.W.
WASHINGTON, D.C. 20591

Sponsor issuing office

KATHLEEN M. FAZEN, ACL-1A
(609)485-4431

FEDERAL AVIATION ADMINISTRATION
TECHNICAL CENTER
OFFICE OF RESEARCH AND TECHNOLOGY
BUILDING 270, ROOM B115
ATLANTIC CITY INT'L AIRPORT,NJ 08405

Security class (U,C,S,TS) : U
ONR resident rep. is ACO (Y/N): N
Defense priority rating :
supplemental sheet
Equipment title vests with: Sponsor X GIT
ITEMS OVER $2,500, IF NOT IN APPROVED BUDGET, REQUIRE PRIOR FAA APPROVAL
Administrative comments -
PROCESSED 6-MO NO-COST EXTENSION

GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

---

Closeout Notice Date 03/28/96

Project No. E-24-X50     Center No. 10/24-6-R8117-0A0

Project Director NEMHAUSER G L     School/Lab ISYE

Sponsor US DEPT OF TRANSPORTATION/FED AVIATION ADMIN

Contract/Grant No. 94-G-016     Contract Entity GTRC

Prime Contract No. ____

Title AIR TRAFFIC CONGESTION DELAY OPTIMIZATION

Effective Completion Date 951031 (Performance) 960131 (Reports)

---

| Closeout Actions Required: | Y/N | Date Submitted |
|---|---|---|
| Final Invoice or Copy of Final Invoice | Y | 960308 |
| Final Report of Inventions and/or Subcontracts | Y | 960319 |
| Government Property Inventory & Related Certificate | Y | |
| Classified Material Certificate | N | |
| Release and Assignment | N | |
| Other ____ | N | |

Comments ____

Subproject Under Main Project No. ____

Continues Project No. ____

Distribution Required:

| | |
|---|---|
| Project Director | Y |
| Administrative Network Representative | Y |
| GTRI Accounting/Grants and Contracts | Y |
| Procurement/Supply Services | Y |
| Research Property Managment | Y |
| Research Security Services | N |
| Reports Coordinator (OCA) | Y |
| GTRC | Y |
| Project File | Y |
| Other ____ | N |
| ____ | N |

NOTE: Final Patent Questionnaire sent to PDPI.

E-24-X50
1

# Air Traffic Congestion Delay Optimization

Six-Month Progress Report

Gregory D. Glockner and George L. Nemhauser

November 30, 1994

## 1 Problem Addressed

The FAA's Air Traffic Control System Command Center (ATCSCC, often called Central Flow) helps coordinate flights nationally when facilities are very congested. If no preventive action is taken, many planes could be forced to wait in airborne queues called holding patterns. Holding patterns are undesirable due to the increased risks and the costs of additional fuel burn. The FAA uses Central Flow to lessen the effects of the airborne holding patterns by changing flight schedules. This research focuses on mathematical optimization tools to find the best possible flow control decisions. This model could be incorporated into a decision support tool for Central Flow.

## 2 Work Completed

### 2.1 Modeling

#### 2.1.1 Structure of the Network Model

We cast the congestion delay optimization problem as a dynamic network flow problem. The network flow paradigm gives the model tremendous flexibility; it is easy to incorporate various factors such as:

- Different kinds of congested facilities
  *e.g.* departure airports, arrival airports, airspace, etc.
- Multiple airports and airspace
- Varied flow control directives
  *e.g.* airborne delay, ground delay, enroute speeding, vectoring

We use a time-space network flow model with stochastic (random) capacities. We represent time using a finite number of time stages. Nodes can represent airports, arrival fixes, intersections of air routes, or any other congested areas at different points in time. The flow in the network represents the air traffic. Travel and delays are represented by directed arcs. When a delay arc has positive flow, it represents airplanes that must be delayed at the corresponding time and location. Other arcs represent travel through congested facilities; these arcs have random capacities. A small example network is given on page 5.

### 2.1.2 Modeling the Random Component

The randomness and the multistage structure of the flow control problem make it very complex. To approximate the randomness, we generate a finite sample of the capacity random variables. This gives us a set of capacity scenarios. Each scenario generates a separate network flow problem.

The complete model is complex due to the coupling constraints that link the scenario subproblems. The coupling constraints are needed to provide "nonanticipativity" among scenarios. To understand nonanticipativity, consider the stochastic decision process. In the first time period, we must make a single decision that applies to all scenarios. The coupling constraints ensure that all first-period decisions are identical. Next, some time passes, and we observe some capacity data. Now we need to make a decision for the second period. We must make a single decision for all scenarios that match our early observations since, at this moment, the scenarios in this group are indistinguishable. Again, the coupling constraints ensure that all decisions in the group are identical. This is repeated for later time periods.

In general, the coupling constraints are used to make the same decision at a point when a pair of scenarios are indistinguishable because of the absence of information that would differentiate them.

## 2.2 Solution Techniques

We want to develop techniques that are very fast for a variety of real-sized problems; speed is essential for real-time implementation.

To test different solution techniques, we have created several test data sets. The test data represent a moderate-sized ground hold problem with varying numbers of capacity scenarios. We have tried to solve the test problems using several algorithmic strategies. We want to refine the different algorithms to find one that will solve large problems in a few minutes of computer time. The following algorithms have been tested:

1. **Simplex method for basic formulation.** All test problems have been solved using several variants of the primal and dual simplex method.
2. **Interior-point method for basic formulation.** All test problems have been solved using several variants of interior point methods.
3. **Basic Dantzig-Wolfe decomposition.** Smaller test problems have been solved using a straightforward implementation of Dantzig-Wolfe decomposition.
4. **Compath decomposition.** Smaller test problems have been solved using *compath decomposition.* Compath decomposition is a technique that we have developed specially for the flow control problem. Compath decomposition exploits the special dynamic network structure to produce special columns for a linear programming master problem. The compath decomposition techniques will be explained in a future paper.

# 3 Work in Progress

We are working on many enhancements to our prototype for the compath decomposition algorithm. Some of these include:

1. **Crash procedure.** To find a better initial feasible solution to the master problem.
2. **Column dropping.** To simplify the master problem by eliminating less useful columns.
3. **Primal-dual algorithm.** To improve dual feasibility of the master problem.
4. **Column selection strategies.** To generate better columns at each iteration of the master problem.

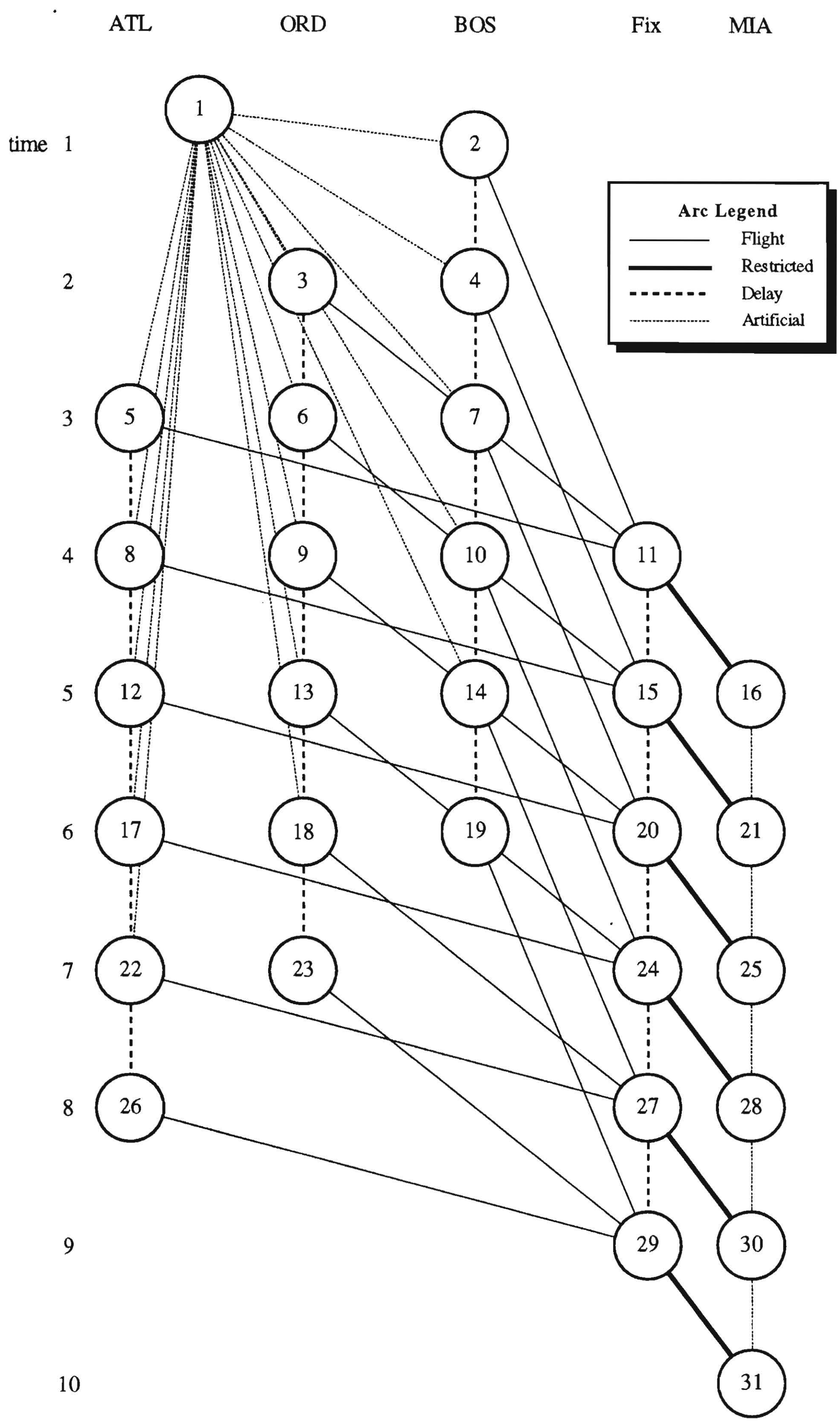
# 4 Future Research

## 4.1 Modeling Issues

Two important theoretical questions remain; both pertain to the generation of the capacity scenarios. First, we need to know how many scenarios are needed for a sufficiently good approximation of the problem. Second, we need to find a good method to sample the scenarios.

## 4.2 Solution Strategies

We will consider alternate ways to solve the basic linear program, e.g. Benders' Decomposition and nonsmooth optimization techniques such as bundle methods. We want to compare the performance of these alternate methods with the methods mentioned in §2.2.




Network Diagram for a Small Multiple-Source Ground Delay Problem

E-24-X50 #2

Georgia Tech

Georgia Institute of Technology

July 31, 1995

Mr. Dave Winer
FAA / ASD-430
800 Independence Ave. SW
Washington, DC 20591

Dear Dave:

This package contains a twelve-month progress report for our project Air Traffic Congestion Delay Optimization. In this past year, we have completed development of our model. This model is too large to be solved quickly using general-purpose linear programming software. Thus, we have also worked on specific algorithmic strategies.

In the upcoming year, we plan to test the model on real-world ground delay situations using the CODAS data set. Also, we intend to refine our algorithmic strategies so that the system will be fast enough to be used operationally in Central Flow.

We hope you are pleased with our current progress. If you have any questions about this research, please do not hesitate to contact me via email at gnemhauser@isye.gatech.edu or via phone at 404-894-2306. I'm sure Greg Glockner will be glad to brief you on this work when he will be in Washington this January at the TRB conference.

We appreciate the FAA support we have received on this project, and we look forward to continuing this FAA research in the coming year.

Very truly yours,

George L. Nemhauser
Institute Professor and Chandler Chair

GLN/amr

xc: File

# Air Traffic Congestion Delay Optimization

Twelve-Month Progress Report

Gregory D. Glockner and George L. Nemhauser

May 22, 1995

## 1 Problem Addressed

The FAA's Air Traffic Control System Command Center (ATCSCC, often called Central Flow) helps coordinate flights nationally when facilities are very congested. If no preventive action is taken, many planes could be forced to wait in airborne queues called holding patterns. Holding patterns are undesirable due to the increased risks and the costs of additional fuel burn. The FAA uses Central Flow to lessen the effects of the airborne holding patterns by changing flight schedules. This research focuses on mathematical optimization tools to find the best possible flow control decisions. This model could be incorporated into a decision support tool for Central Flow.

Our research may be divided into three main components: model development, methodology, and output validation. Model development covers the formulation of the model framework that describes air traffic flow control. The robustness of our model makes it quite complex, requiring specialized mathematical techniques and algorithms to solve it. These mathematical techniques are the solution methodology. Finally, the output validation tests both the quality of the model and the effectiveness of our specialized mathematical techniques.

## 2 Completed Research

### 2.1 Model Development

In this research, we have developed a *model framework* rather than a specific model. Our model framework can describe congested areas, flight opportunities, and delay opportunities. This approach is simple yet robust: the policy choices within the FAA can determine the various delay opportunities and their associated costs. This gives our model tremendous flexibility; it could be used with current air traffic control policies or could be used with future "free-flight" policies.

Our development of the model framework was completed during the first year of this grant. Our approach to air traffic congestion delay optimization is based on a dynamic network flow model where some arcs have uncertain capacity. These special arcs capture the uncertainty in predicting precise future airport or airspace acceptance rates. The resulting solution can "hedge" against multiple capacity predictions, giving a very robust solution.

Recall that a network flow model is a generalization of an assignment model, so this can incorporate the assignment models in [6], [10], and [9]. Dynamic network flow models are described in §III.9 of [4] and §19.6 of [1]. Suppose we have a set $L = \{1, \ldots, \ell\}$ of locations and discrete time intervals from 1 to $T$. Some of these sites are congested, which can represent an arrival airport, a departure airport, or some section of enroute airspace. For a congested site $\ell_i$, we define two locations $\ell_i'$ and $\ell_i''$ that correspond to entering and leaving the congested site. We create a set of nodes $N = L \times T$, where each node $n_j = (\ell_j, t_j)$ refers to a location $\ell_j$ at time $t_j$. The arcs $a \in A$ in the directed graph $G = (N, A)$ have the form $a = \langle(\ell_{i_1}, t_{j_1}), (\ell_{i_2}, t_{j_2})\rangle$ and $t_{j_2} > t_{j_1}$ for all $a \in A$.

There are five kinds of arcs in this directed graph. First, we define a flight arc from node $(\ell_{i_1}, t_{j_1})$ to node $(\ell_{i_2}, t_{j_2})$ if there is some flight from location $\ell_{i_1}$ to $\ell_{i_2}$ that takes time $t_{j_2} - t_{j_1}$. Second, we define a delay arc from node $(\ell_i, t_j)$ to node $(\ell_i, t_j + 1)$ if a plane can take delay at location $\ell_i$. Third, we define a restricted arc for congested site $i$ as an arc from node $(\ell_i', t_j)$ to node $(\ell_i'', t_j + 1)$. The restricted arcs are the special arcs with uncertain capacity. Fourth, we define sink arcs from node $(\ell_i, t_j)$ to node $(\ell_i, T + 1)$ if $\ell_i$ is a destination location. Fifth, we define an overflow arc from node $(\ell_{i_1}, T)$ to node $(\ell_{i_2}, T + 1)$. The overflow arcs are used when a plane is delayed beyond the congestion period and flown extremely late (for example, departing at 2 a.m.). The restricted arcs will have finite capacity to model the congestion at location $i$; the flight arcs, delay arcs, sink arcs, and overflow arcs have infinite capacity. Similarly, the delay arcs have an associated delay cost; the flight arcs, restricted arcs, and sink arcs have zero cost. The overflow arcs may have an associated cost. The five kinds of arcs are summarized in Table 1.

| Arc Type | Form | Cost | Capacity | Notes |
|---|---|---|---|---|
| Flight | $\langle(\ell_{i_1}, t_{j_1}), (\ell_{i_2}, t_{j_2})\rangle$ | 0 | Infinite | $\ell_{i_1} \neq \ell_{i_2}$, $t_{j_2} > t_{j_1}$ |
| Delay | $\langle(\ell_i, t_j), (\ell_i, t_j + 1)\rangle$ | Usually positive | Infinite | |
| Restricted | $\langle(\ell_i', t_j), (\ell_i'', t_j + 1)\rangle$ | 0 | Positive | May be random |
| Sink | $\langle(\ell_i, t_j), (\ell_i, T + 1)\rangle$ | 0 | Infinite | $\ell_i$ is a sink location |
| Overflow | $\langle(\ell_{i_1}, T), (\ell_{i_2}, T + 1)\rangle$ | Varies | Infinite | $\ell_{i_1} \neq \ell_{i_2}$ |

Table 1: Summary of Arc Types

The flights are represented as flow in the network. Each flight corresponds to a commodity (source-sink pair). A source node corresponds to a flight's starting location and departure time. Each commodity has a specific ending location, but it may have several different candidates for ending times. Thus, for each ending location $\ell_e$, we designate a supersink node $(\ell_e, T+1)$ and use the artificial sink arcs $\langle(\ell_e, t), (\ell_e, T+1)\rangle$ for $1 \leq t \leq T$. These sink arcs have infinite capacity and zero cost. In addition, we use infinite capacity overflow arcs $\langle(\ell_s, T), (\ell_e, T+1)\rangle$ for each starting-ending location pair $\ell_s, \ell_e$. The combination of overflow and delay arcs ensure that every problem has a feasible flow regardless of the random capacities.

A small example can be found in Figure 1. In this example, there are flights departing from nodes (D1, 3), (D1, 4), (D1, 5), (D1, 6), (D1, 7), (D2, 2), (D2, 3), (D2, 4), (D2, 5), (D2, 6), (D3, 1), (D3, 2), (D3, 3), (D3, 4), (D3, 5). Note that nodes (D1, 8), (D2, 7), (D3, 6) are not supply nodes but rather late departure nodes. These nodes along with the overflow arcs correspond to the situation when a flight is delayed far beyond the congested period. The supersink is the square node; this node is actually (A$''$, 10), but it appears in the center to improve the clarity of the picture.

Some other aspects may be incorporated into the model. For instance, several flights may be grouped together as one commodity. Also, the sink or overflow arcs could have nonzero "penalty" costs to prevent a severe delay on any particular flight. Enroute speeding or vectoring can be incorporated using additional flight arcs that have infinite capacity and nonzero costs. Furthermore, one can add a set of additional "fairness criteria" in order to balance the delays among the different users of the air traffic control system. This network flow paradigm gives tremendous flexibility to describe these and many other phenomena.

## 2.2 Mathematical Methodology

This was the other main research focus in the first year of this grant. A principal goal of our methodological research is to be able to solve realistic flow management problems within twenty minutes on a powerful Unix workstation. This will allow the FAA to use our robust modeling techniques and get solutions reasonably quickly without extraordinary investment in computer hardware.

We describe the uncertainty using capacity scenarios, where each scenario represents a possible collection of capacities. Scenario analysis was first described in [3]; a recent reference may be found in [7]. In the dynalock-angular structure. For example, a problem with two scenarios has the

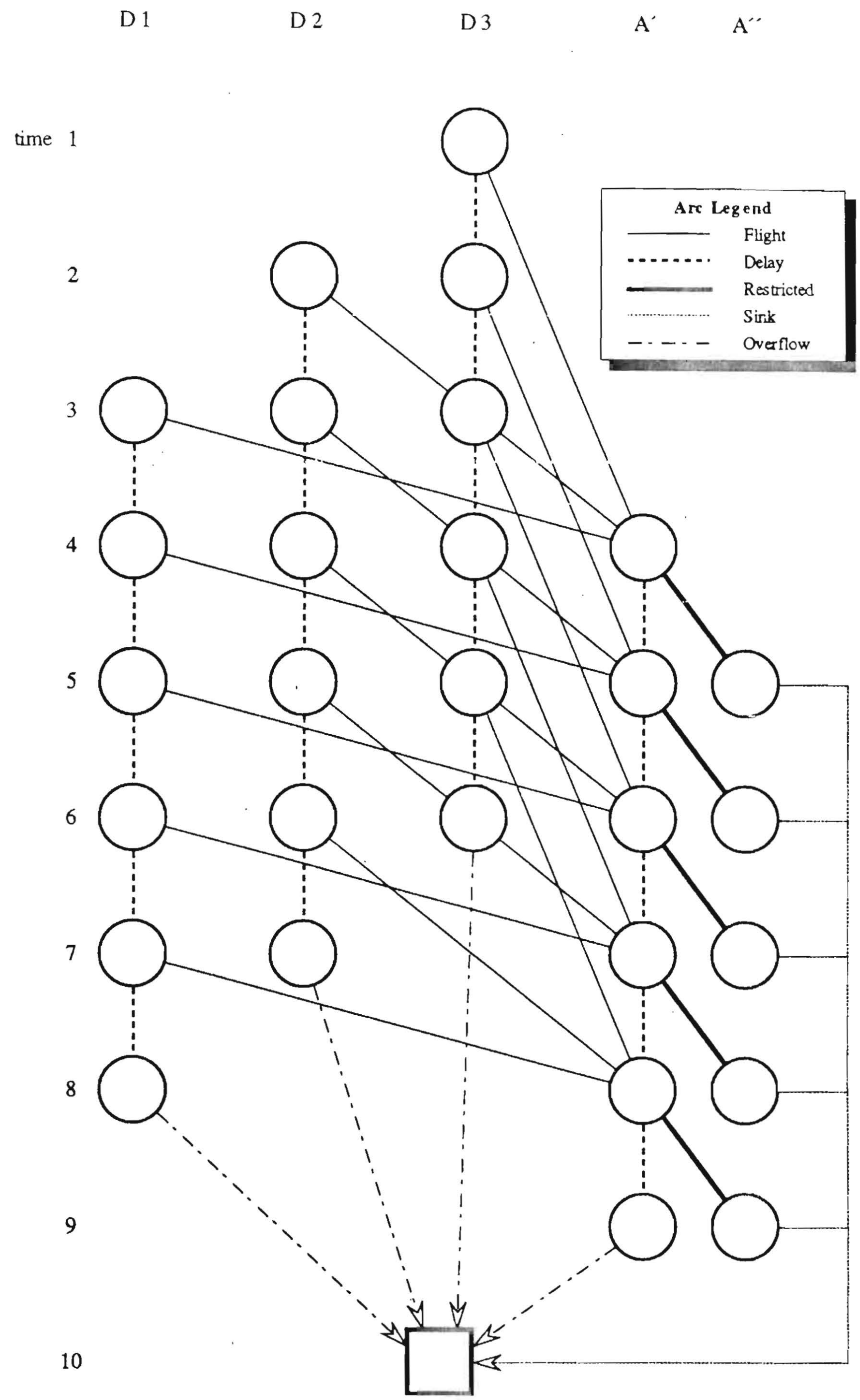



Figure 1: Diagram of Flow Management Problem as a Dynamic Network Flow Problem

following form:

$$
\begin{array}{lccccccccl}
\min & c_x x_1 & + & c_y y_1 & + & c_x x_2 & + & c_y y_2 & & \\
\text{s.t.} & & & & & & & & & \\
& N_x x_1 & + & N_y y_1 & & & & & = & b \\
& & & & & N_x x_2 & + & N_y y_2 & = & b \\
& x_1 & & & & & & & \leq & K_x \\
& & & y_1 & & & & & \leq & K_{y_1} \\
& & & & & x_2 & & & \leq & K_x \\
& & & & & & & y_2 & \leq & K_{y_2} \\
& x_1 & & & - & x_2 & & & = & 0 \\
& & & & & & & x_1, y_1, x_2, y_2 & \geq & 0
\end{array}
$$

We solved a multiple scenario test problem using several general techniques, but the runtimes were quite lengthy. We have solved a test problem naively using commercial software (CPLEX and OSL). The run times using several variants of the simplex method (*e.g.* steepest edge, dual simplex, etc.) were just fair except for small problems. The performance of interior point methods on the dual of the formulation has been good. However, the memory requirements for the factorizations grew enormous; we were unable to try a 1024 scenario test problem. Furthermore, the time to recover a basis was very high, which essentially erased the advantage of the interior point algorithm. All our test problems so far had a single commodity with several source nodes, which is appropriate for a ground delays problem.

The linear program has very special structure: it has a block angular form, where each block is an acyclic directed network flow subproblem. We spent considerable time exploring this special structure. First, we experimented with linear programming decomposition techniques. We implemented the "natural" form of Dantzig-Wolfe decomposition using OSL's built-in decomposition routines. The time to converge to the optimal solution was terrible, and we ran out of memory trying to get a solution for a medium-sized test problem. We also studied some problem-specific decomposition strategies. An unusual implementation of Dantzig-Wolfe decomposition led to a special column generation strategy which we called *compath decomposition.* In this problem, a compath represents flow management decisions over time for a particular flight. In general, a compath is a collection of paths, one for each scenario, that is logically consistent with respect to the scenarios. Logical consistency requires that whenever two scenarios appear to be identical, then all decisions for these scenarios up to that point must also be identical. Compath decomposition is similar to path decomposition techniques for network network flow problems. By using the special structure in this dynamic network, we were able to generate solutions much more quickly than naive algorithms.

We have tested different refinements of compath decomposition. In our compath decomposition code, we solve the master problem using commercial linear programming software (CPLEX), and we solve the column generation by a compath generation routine written in C. We developed some simple preprocessing techniques to eliminate redundant rows in the master problem, which improved the dual degeneracy associated with the master problem. We also wrote an initialization routine that exploits the compath structure to generate a good initial feasible solution. When we incorporated these techniques along with a primal-dual techniques, the compath decomposition algorithm solved all our smaller test problems very quickly. Unfortunately, it failed to solve our largest test problem (1024 scenarios) quickly. Part of the current difficulty with the largest test problem is that the time to compute an optimal solution to the master problem grows quite large.

### 2.3 Output Validation

This was not part of our original research proposal, and we have not started this work. However, we want to perform empirical tests on the quality of the solutions obtained from our model. We want to estimate the potential reduction in delay costs by our flow management algorithms as compared to ATCSCC's current first-come-first-served policy, also known as Groverjack. We may also compare our model with other heuristics ([2], [8], and [5]), and we may also compare our model with more simple models, such as the deterministic model in [10] and the stochastic ground delay models in [6] and [9]. We will solve several test problems and compare their net results via a simulation. At the same time, it may be possible to use the simulation to compare sampling strategies and other approximation issues related to this research.

## 3 Summary

In the past year under this grant, we have developed a suitable model framework. We have performed extensive computational tests on the model framework. The computation times have been fast for the small problems, but not yet fast enough for the larger problems. In the future, we plan to continue studying the special structure of this problem. We hope that more insight into the special structure of this problem will lead to significant reductions in computation time. Specifically, several recent observations about the polyhedral structure of the linear program may have tremendous applicability to a new computational algorithm. Also, we plan to perform some output analysis to compare the potential benefits of these more complex flow management optimization models.

## References


[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs NJ, 1993.

[2] G. Andreatta and G. Romanin-Jacur. Aircraft Flow Management Under Congestion. *Transportation Science* 21, 249–253, 1987.

[3] G. B. Dantzig. Linear Programming under Uncertainty. *Management Science* 1, 197–206, 1955.

[4] L. R. Ford, Jr. and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.

[5] G. D. Glockner. MIDCORT: Minimizing Delay Costs in Real Time. Technical report, Federal Aviation Administration AOR-200, Washington DC, 1992.

[6] O. Richetta and A. R. Odoni. Dynamic Solution to the Ground-Holding Problem in Air Traffic Control. *Transportation Research Part A: Policy and Practice* 28, 167–185, 1994.

[7] R. T. Rockafellar and R. J.-B. Wets. Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research* 16, 119–147, 1991.

[8] M. Terrab and A. R. Odoni. Strategic Flow Management for Air Traffic Control. *Operations Research* 41, 138–152, 1993.

[9] P. B. Vranas, D. J. Bertsimas, and A. R. Odoni. Dynamic Ground-Holding Policies for a Network of Airports. *Transportation Science* 28, 275–291, 1994.

[10] P. B. Vranas, D. J. Bertsimas, and A. R. Odoni. The Multi-Airport Ground-Holding Problem in Air Traffic Control. *Operations Research* 42, 249–261, 1994.

E-24-X50 #'s 3, 4

US Department of Transportation
**Federal Aviation Administration**

# FINAL PROJECT REPORT

Form Approved:
O.M.B. No. 2120-0559

## PART I - PROJECT IDENTIFICATION INFORMATION

| | | |
|---|---|---|
| 1. Institution and Address<br>Georgia Institute of Technology<br>Atlanta, Georgia 30332 | 2. FAA Program<br><br>4. Award Period<br>From To | 3. FAA Award Number<br><br>5. Cumulative Award Amount |

6. Project Title

Air Traffic Congestion Delay Optimization

## SUMMARY OF COMPLETED PROJECT (For Public Use)

## PART III - TECHNICAL INFORMATION (For Program Management Uses)

| 1. ITEM (Check appropriate blocks) | NONE | ATTACHED | PREVIOUSLY FURNISHED | TO BE FURNISHED SEPARATELY TO PROGRAM: Check (✓) | Approx. Date |
|---|---|---|---|---|---|
| a. Abstracts of Theses | | | | X | 6/30/96 |
| b. Publication Citations | | X | | | |
| c. Data on Scientific Collaborators | | X | | | |
| d. Information on Inventions | X | | | | |
| e. Technical Description of Project and Results | | X | | | |
| f. Other (specify) | | | | | |

| | | |
|---|---|---|
| 2. Principal Investigator / Project Director Name (Typed)<br>George L. Nemhauser | 3. Principal Investigator / Project Director Signature | 4. Date<br>3/15/96 |

## Abstract

The FAA's Air Traffic Control System Command Center (ATCSCC, often called Central Flow) helps coordinate flights nationally when facilities are very congested. If no preventive action is taken, many planes could be forced to wait in airborne queues called holding patterns. Holding patterns are undesirable due to the increased risks and the additional fuel burn. The FAA uses Central Flow to lessen the effects of the airborne holding patterns by changing flight schedules. This research focuses on mathematical optimization tools to find the best possible flow control decisions. This model could be incorporated into a decision support tool for Central Flow. The research has three components: model development, methodology, and output validation. Model development covers the formulation of the model framework that describes air traffic flow control. The robustness of our model makes it quite complex, requiring specialized mathematical methodólogy to solve it. Finally, the output validation tests both the quality of the model and the effectiveness of our specialized mathematical techniques. During the time of this grant, we completed one paper that describes describes the model and output validation. We also worked on methodology, which will appear in two other papers.

## Publication Citations

[1] G. D. Glockner. Effects of Air Traffic Congestion Delays Under Several Flow Management Policies. Report 96-01, Logistics Engineering Center, School of Industrial and Systems Engineering, Georgia Institute of Technology. To appear in *Transportation Research Record.*

[2] G. D. Glockner and G. L. Nemhauser. Dynamic Network Flow with Uncertain Arc Capacities: Formulation and Problem Structure. To appear.

[3] G. D. Glockner and G. L. Nemhauser. Dynamic Network Flow with Uncertain Arc Capacities: Algorithms and Computational Results. To appear.

## Data on Scientific Collaborators

Greg Glockner
PhD candidate
PhD expected December 1996

# Air Traffic Congestion Delay Optimization

Technical Summary

March 12, 1996

## 1 Problem Addressed

The FAA's Air Traffic Control System Command Center (ATCSCC, often called Central Flow) helps coordinate flights nationally when facilities are very congested. If no preventive action is taken, many planes could be forced to wait in airborne queues called holding patterns. Holding patterns are undesirable due to the increased risks and the costs of additional fuel burn. The FAA uses Central Flow to lessen the effects of the airborne holding patterns by changing flight schedules. This research focuses on mathematical optimization tools to find the best possible flow control decisions. This model could be incorporated into a decision support tool for Central Flow.

Our research may be divided into three main components: model development, methodology, and output validation. Model development covers the formulation of the model framework that describes air traffic flow control. The robustness of our model makes it quite complex, requiring specialized mathematical methodology to solve it. Finally, the output validation tests both the quality of the model and the effectiveness of our specialized mathematical techniques. During the time of this grant, we completed [6], which describes the model and output validation. We have also worked on methodology, which will appear in [7].

## 2 Completed Research

### 2.1 Model Development

In this research, we have developed a *model framework* rather than a specific model. Our model framework can describe congested areas, flight opportunities, and delay opportunities. This approach is simple yet robust: the policy choices within the FAA can determine the various delay opportunities and their associated costs. This gives our model tremendous flexibility; it could be used with current air traffic control policies or with free-flight policies in the future.

The development of the model framework was completed during the time of this grant. Our approach to air traffic congestion delay optimization is based on a dynamic network flow model where some arcs have uncertain capacity. These special arcs capture the uncertainty in predicting precise future airport or airspace acceptance rates. The resulting solution can "hedge" against multiple capacity predictions, giving a very robust solution.

Recall that a network flow model is a generalization of an assignment model, so this can incorporate the assignment models in [8], [12], and [11]. Dynamic network flow models are described in §III.9 of [4] and §19.6 of [1]. Suppose we have a set $L = \{1, \ldots, \ell\}$ of locations and discrete time intervals from 1 to $T$. Some of these sites are congested, which can represent an arrival airport, a departure airport, or some section of enroute airspace. For a congested site $\ell_i$, we define two locations $\ell_i'$ and $\ell_i''$ that correspond to entering and leaving the congested site. We create a set of nodes $N = L \times T$, where each node $n_j = (\ell_j, t_j)$ refers to a location $\ell_j$ at time $t_j$. The arcs $a \in A$ in the directed graph $G = (N, A)$ have the form $a = \langle(\ell_{i_1}, t_{j_1}), (\ell_{i_2}, t_{j_2})\rangle$ and $t_{j_2} > t_{j_1}$ for all $a \in A$.

There are five kinds of arcs in this directed graph. First, we define a flight arc from node $(\ell_{i_1}, t_{j_1})$ to node $(\ell_{i_2}, t_{j_2})$ if there is some flight from location $\ell_{i_1}$ to $\ell_{i_2}$ that takes time $t_{j_2} - t_{j_1}$. Second, we define a delay arc from node $(\ell_i, t_j)$ to node $(\ell_i, t_j + 1)$ if a plane can take delay at location $\ell_i$. Third, we define a restricted arc for congested site $i$ as an arc from node $(\ell_i', t_j)$ to node $(\ell_i'', t_j + 1)$. The restricted arcs are the special arcs with uncertain capacity. Fourth, we define sink arcs from node $(\ell_i, t_j)$ to node $(\ell_i, T + 1)$ if $\ell_i$ is a destination location. Fifth, we define an overflow arc from node $(\ell_{i_1}, T)$ to node $(\ell_{i_2}, T + 1)$. The overflow arcs are used when a plane is delayed beyond the congestion period and flown extremely late (for example, departing at 2 a.m.). The restricted arcs will have finite capacity to model the congestion at location $i$; the flight arcs, delay arcs, sink arcs, and overflow arcs have infinite capacity. Similarly, the delay arcs have an associated delay cost; the flight arcs, restricted arcs, and sink arcs have zero cost. The overflow arcs may have an associated cost. The five kinds of arcs are summarized in Table 1.

| **Arc Type** | **Form** | **Cost** | **Capacity** | **Notes** |
|---|---|---|---|---|
| Flight | $\langle(\ell_{i_1}, t_{j_1}), (\ell_{i_2}, t_{j_2})\rangle$ | 0 | Infinite | $\ell_{i_1} \neq \ell_{i_2}$, $t_{j_2} > t_{j_1}$ |
| Delay | $\langle(\ell_i, t_j), (\ell_i, t_j + 1)\rangle$ | Usually positive | Infinite | |
| Restricted | $\langle(\ell_i', t_j), (\ell_i'', t_j + 1)\rangle$ | 0 | Positive | May be random |
| Sink | $\langle(\ell_i, t_j), (\ell_i, T + 1)\rangle$ | 0 | Infinite | $\ell_i$ is a sink location |
| Overflow | $\langle(\ell_{i_1}, T), (\ell_{i_2}, T + 1)\rangle$ | Varies | Infinite | $\ell_{i_1} \neq \ell_{i_2}$ |

Table 1: Summary of Arc Types

The flights are represented by flow in the network. Each flight corresponds to a commodity (source-sink pair). A source node corresponds to a flight's starting location and departure time. Each commodity has a specific ending location, but it may have several different candidates for ending times. Thus, for each ending location $\ell_e$, we designate a supersink node $(\ell_e, T+1)$ and use the artificial sink arcs $\langle(\ell_e, t), (\ell_e, T+1)\rangle$ for $1 \leq t \leq T$. These sink arcs have infinite capacity and zero cost. In addition, we use infinite capacity overflow arcs $\langle(\ell_s, T), (\ell_e, T+1)\rangle$ for each starting-ending location pair $\ell_s, \ell_e$. The combination of overflow and delay arcs ensure that every problem has a feasible flow regardless of the random capacities.

A small example can be found in Figure 1. In this example, there are flights departing from nodes (D1, 3), (D1, 4), (D1, 5), (D1, 6), (D1, 7), (D2, 2), (D2, 3), (D2, 4), (D2, 5), (D2, 6), (D3, 1), (D3, 2), (D3, 3), (D3, 4), (D3, 5). Note that nodes (D1, 8), (D2, 7), (D3, 6) are not supply nodes but rather late departure nodes. These nodes along with the overflow arcs correspond to the situation when a flight is delayed far beyond the congested period. The supersink is the square node; this node is actually (A$''$, 10), but it appears in the center to improve the clarity of the picture.

Other realistic aspects may be incorporated into the model. First, the sink or overflow arcs can have nonzero "penalty" costs to prevent a severe delay on any particular flight. This can model downline effects: to avoid delaying one flight so long that later flights flown by that crew or airplane are also delayed. Second, dependent delay effects can be modeled by multiple locations in the network model. Third, enroute speeding or vectoring can be incorporated using additional flight arcs that have infinite capacity and nonzero costs. Fourth, one can limit the number of planes allowed in holding patterns by putting a capacity restriction on the corresponding delay arc. Fifth, one can add a set of additional "fairness criteria" in order to balance the delays among the different users of the air traffic control system. Sixth, this model could be used to optimize the controlled arrival times under a free-flight system. These realistic aspects are described in detail in [6].

### 2.2 Mathematical Methodology

This was another main research focus of this grant. A principal goal of our methodological research was to be able to solve realistic flow management problems within twenty minutes on a powerful Unix workstation. This will allow the FAA to use our robust modeling techniques and get solutions quickly without extraordinary investment in computer hardware.

We describe the uncertainty using capacity scenarios, where each scenario represents a possible collection of capacities. Scenario analysis was first described in [3]; a recent reference may be found in [9]. In the dynamic network flow problem, each scenario induces a network flow subproblem.

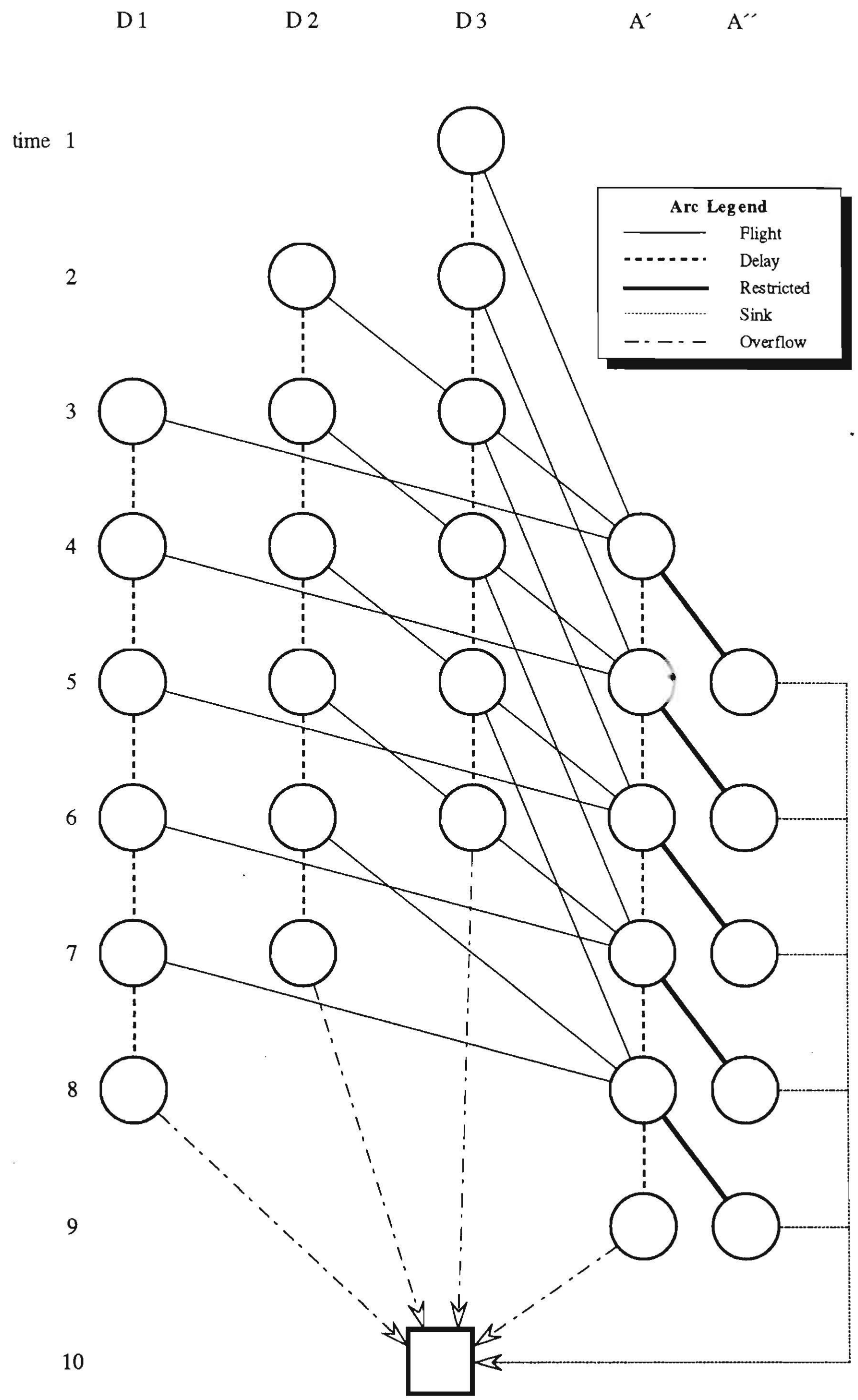



Figure 1: Diagram of Flow Management Problem as a Dynamic Network Flow Problem

The overall problem has side constraints to link the scenarios. These constraints give the problem block-angular structure. Here is the linear programming relaxation for our problem:

$$\min \quad \sum_k p_k(c^k \cdot x^k)$$

s.t.

$$\begin{array}{rclll} Nx^k & = & b & \forall k & (1) \\ 0 \leq x_{ij}^k & \leq & \bar{\omega}_{ij}^k & \forall k; \forall (i,j) \in \mathcal{A} & (2) \\ x_{ij}^{k'} - x_{ij}^{k''} & = & 0 & \forall (i,j) \in \mathcal{A}; \forall k', k'' : \bar{\omega}_{\lfloor t(i) \rfloor}^{k'} = \bar{\omega}_{\lfloor t(i) \rfloor}^{k''} & (3) \end{array}$$

The constraints (1) are the flow conservation constraints for each scenario, and the constraints (2) are the uncertain arc capacities. The constraints (3) are linking constraints, used to ensure compatibility between the scenario subproblems.

We solved a multiple scenario test problem using several general techniques, but the runtimes were quite lengthy. We have solved a test problem naively using commercial software (CPLEX and OSL). The run times using several variants of the simplex method (*e.g.* steepest edge, dual simplex, etc.) were just fair except for small problems. The performance of interior point methods on the dual of the formulation has been good. However, the memory requirements for the factorizations grew enormous; we were unable to try a 1024 scenario test problem. Furthermore, the time to recover a basis was very high, which essentially erased the advantage of the interior point algorithm. All our test problems so far had a single commodity with several source nodes, which is appropriate for a ground delays problem.

This linear program has very special structure. We spent considerable time exploring this special structure. First, it has a block angular form, where each block is an acyclic directed network flow subproblem. We experimented with exploiting the block-angular structure using OSL's built-in decomposition routines. The time to converge to the optimal solution was terrible, and we ran out of memory trying to get a solution for a medium-sized test problem. Second, we also studied some problem-specific decomposition strategies. The structure of the dynamic network flow matrix suggested a new decomposition strategy, which we call *compath decomposition.* Compath decomposition is described briefly in [6] and in detail in [7]. In this problem, a compath represents flow management decisions over time for a particular flight. In general, a compath is a collection of paths, one for each scenario, that is logically consistent with respect to the scenarios. Logical consistency requires that whenever two scenarios appear to be identical, then all decisions for these scenarios up to that point must also be identical. Compath decomposition is similar to path decomposition techniques for network network flow problems. By using the special structure in this dynamic network, we are able to generate solutions much more quickly than naive algorithms.

We have tested different refinements of compath decomposition. In our compath decomposition code, we solve the master problem using commercial linear programming software (CPLEX), and we solve the column generation by a compath generation routine written in C. We developed some simple preprocessing techniques to eliminate redundant rows in the master problem, which improved the dual degeneracy associated with the master problem. We also wrote an initialization routine that exploits the compath structure to generate a good initial feasible solution. When we incorporated these techniques along with a primal-dual techniques, the compath decomposition algorithm solved all our smaller test problems very quickly. In the future, we plan to investigate Lagrangian decomposition techniques and techniques from nonsmooth optimization.

## 2.3 Output Validation

This was not part of our original research proposal, but we found it to be useful to evaluate the performance of our model in real-world air traffic management. This was an important test since any flow management model generally finds an optimal solution to a simplified form of the problem, which may not be a low-cost solution for the actual problem. We also used these tests to compare the performance of our model against other models for air traffic flow management. There were two parts to this research: solution generation and simulation. In the solution generation stage, the different test problems were used as inputs to the various flow management models. After generating the flow management solutions from the different models, we used a statistical simulation to estimate the real performance of the solutions. This procedure is described in detail in [6].

We studied five airports: ATL, DCA, DEN, MCO, SEA. The airports were selected because they have had historically large amounts of congestion delays, and because they have different kinds of operations. We used CODAS data to get desired flight plans. While CODAS data do not include many flights, they are free of many of the biases found in other data such as ETMS or OAG. Since we were assessing the relative performance of our model, we expect that CODAS will give reliable comparisons. Cost data were obtained by estimates from the Air Transport Organization. Again, inherent biases in ATA data will not affect our relative results. These results are described in [6], but we repeat the summary here:

| Problem | Expected Savings of MSDNF Over Other Models: In $ FCFS | NFC | In % FCFS | NFC |
|---|---|---|---|---|
| ATL | 8,910 | 16,863 | 20.3 | 32.5 |
| DCA | 8,235 | -2,476 | 4.2 | -1.3 |
| DEN | 3,060 | 10,710 | 2.4 | 7.8 |
| MCO | 376 | 1,153 | 7.4 | 19.6 |
| SEA | 19,546 | 61,264 | 11.1 | 28.1 |
| Average | | | 9.1 | 17.3 |

In this table, FCFS represents current FAA first-come, first served flow management policies, and NFC represents no flow control, which allows all flights to depart on time.

## 3 Summary

Under this grant, we have developed a flexible model framework for flow management. This model framework could be used to model realistic flow management decisions. In addition, we have tested the model framework computationally. In the future, we plan to continue studying the special structure of this problem to reduce computation time further. Improved understanding of the polyhedral structure of the linear program should improve the performance of compath decomposition. Finally, we performed output analysis that demonstrated that our model has the potential to give a large reduction in air traffic congestion delay effects. These savings suggest that it is worthwhile to continue developing algorithms so that this model can be used operationally.

## References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs NJ, 1993.

[2] G. Andreatta and G. Romanin-Jacur. Aircraft Flow Management Under Congestion. *Transportation Science* 21, 249–253, 1987.

[3] G. B. Dantzig. Linear Programming under Uncertainty. *Management Science* 1, 197–206, 1955.

[4] L. R. Ford, Jr. and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton NJ, 1962.

[5] G. D. Glockner. MIDCORT: Minimizing Delay Costs in Real Time. Technical report, Federal Aviation Administration AOR-200, Washington DC, 1992.

[6] G. D. Glockner. Effects of Air Traffic Congestion Delays Under Several Flow Management Policies. Report 96-01, Logistics Engineering Center, School of Industrial and Systems Engineering, Georgia Institute of Technology. To appear in *Transportation Research Record.*

[7] G. D. Glockner and G. L. Nemhauser. Dynamic Network Flow with Uncertain Arc Capacities: Formulation and Problem Structure. To appear.

[8] O. Richetta and A. R. Odoni. Dynamic Solution to the Ground-Holding Problem in Air Traffic Control. *Transportation Research Part A: Policy and Practice* 28, 167–185, 1994.

[9] R. T. Rockafellar and R. J.-B. Wets. Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research* 16, 119–147, 1991.

[10] M. Terrab and A. R. Odoni. Strategic Flow Management for Air Traffic Control. *Operations Research* 41, 138–152, 1993.

[11] P. B. Vranas, D. J. Bertsimas, and A. R. Odoni. Dynamic Ground-Holding Policies for a Network of Airports. *Transportation Science* 28, 275–291, 1994.

[12] P. B. Vranas, D. J. Bertsimas, and A. R. Odoni. The Multi-Airport Ground-Holding Problem in Air Traffic Control. *Operations Research* 42, 249–261, 1994.